

DETECTION OF DEFERRED LUGGAGE IN PUBLIC AREAS

Roman Ligocki

Bachelor, VUT FEEC

E-mail: xligoc02@vutbr.cz

Supervised by: Václav Uher

E-mail: vaclav.uher@vutbr.cz

Abstract: Goal of this paper is to describe created program for detection of deferred luggage in public areas, that could be possibly dangerous for their surrounding. Text describes usage of neural network YOLO for object detection from digital image. Paper also contains description of used object tracking algorithm and formula for image comparison. Algorithm for object tracking, has been implemented based on image comparison and distance between objects. Result of this project is application, that is able to detect deferred luggage.

Keywords: Deferred luggage, Object detection, Neural network, Security

1 TEORETICKÝ ÚVOD

V dnešních časech se velmi často mluví o teroristických útocích, které jsou páchany na veřejných prostorech. Oběti v tomto případě jsou civilní osoby. Problémem je, že kromě detekce a následné neutralizace teroristické složky pomocí tajných služeb, je jedinou možností zachytit pachatele před spáchaním teroristického útoku. Z důvodu, že pachatel většinou chce spáchaný zločin přežít, je nutné aby útok byl proveden nepřímo. Příkladem může být atentát pomocí výbušné bomby ukryté v cestovním kufří. Tato práce se věnuje návrhu programu, který bude schopen na základě snímku z bezpečnostní kamery detekovat opuštěná zavazadla, která mohou být nebezpečná svému okolí. Program byl implementován v programovacím jazyce Python

2 POPIS VSTUPNÍCH DAT

Pro návrh programu bylo použito video z soutěže, kde účastníci měli za úkol navrhnout program schopný detekovat odložená zavazadla. Pro účely soutěže vznikla sada videí, kde jsou nasimulovány různé případy manipulace se zavazadlem. Různorodost videí umožňuje testovat robustnost algoritmu. Může se jednat o velký dav, který s velkou frekvencí prochází před zavazadlem. Algoritmus v takovém případě musí být dostatečně robustní a podezřelá zavazadla i při nepříznivých podmínkách detekovat.

3 POPIS VLASTNÍHO ŘEŠENÍ

Pro zachycení obrazu z veřejného prostoru je nutné použít kameru. Výstupem z této kamery je nejčastěji video sekvence poskládaná z třiceti nebo šedesáti snímků za sekundu. Každý snímek po nahrání do paměti pomocí knihovny je reprezentován tensorem, tedy polem se třemi rozměry. Dva rozměry jsou v tomto případě výška a šířka obrazu v pixelech. Třetí rozměr jsou tři barevné vrstvy. Na těchto snímcích se pohybují osoby a zavazadla, které program následně má za cíl detekovat a vyhledat mezi nimi podezřelá zavazadla

Pro řešení tohoto problému došlo k návrhu a implementaci programu. Program se skládá ze tří částí:

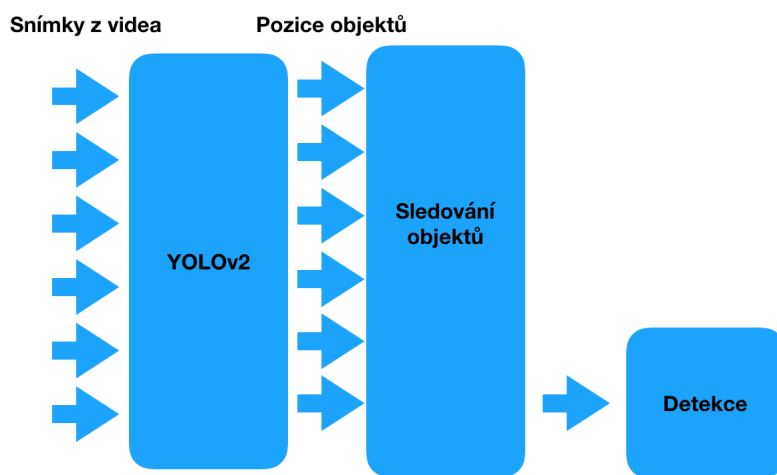
zachycení snímku z video sekvence, detekce osob a zavazadel v obrazu a detekce odloženého zavazadla

V první části programu se používá framework OpenCV, který slouží pro práci s obrazem. Z videa se sekvenčně vytahují snímky, které se ukládají do paměti v podobě tenzoru. V této části programu rovněž dochází ke změně rozlišení obrazu dle vstupu neuronové sítě.

Druhá část programu je složená z neuronové sítě YOLO, která detekuje objekty na snímcích a python wrapperu pro ovládání této neuronové sítě. Z důvodu, že neuronové sítě jsou výpočetně náročné, bylo nutné pro jejich akceleraci použít grafické jednotky s integrovanými CUDA procesory. Vstupem do wrapperu ovládajícího YOLO síť je snímek s rozlišením 488x488 pixelů. Výstupem jsou pozice objektu a jejich klasifikace, zda se jedná o člověka nebo zavazadlo. Pozice jsou dále použita pro vytvoření snímku, kde se zobrazují detekované objekty a jejich klasifikace, nebo pro algoritmus sledující objekty v čase.

Pro sledování objektů mezi snímky byl navržen vlastní algoritmus na základě porovnávání výstřžků objektů z aktuálního snímku s výstřžky objektů z předchozího snímku. Rozdíl je vypočten pomocí střední kvadratické odchylky (MSE), která se počítá pro každou barevnou vrstvu z každého pixelu snímku. Pro výpočet střední kvadratické odchylky se používá vzorec 1. Vzniklá chyba se násobí vzdáleností objektů mezi sebou. Objekty, mezi kterými vznikla nejmenší chyba a zároveň nepřásáhla pevně stanovenou hodnotu, jsou považovány za stejný objekt.

$$MSE = \frac{1}{MN} \sum_{m=1}^{i=0} \sum_{n=1}^{j=0} [I(i, j) - K(i, j)]^2 \quad (1)$$



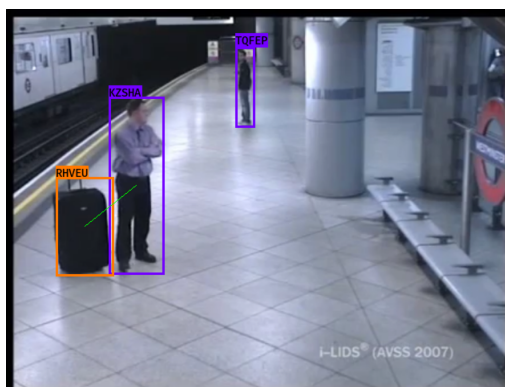
Obrázek 1: Grafické znázornění struktury programu

Program rovněž ukládá informace o pohybu objektů, což umožňuje zjistit podobnost křivek pohybů objektů. Na základě podobnosti pohybu se pak vyhodnocuje, zda zavazadla mají vlastníka mezi osobami na snímku. Pokud zavazadlo nemá svého majitele a nepohybuje se, zařadí se zavazadlo mezi podezřelá a dojde k jeho upozornění na výstupním videu pomocí červeného čtverce. Vzorec 2 slouží k výpočtu podobnosti křivek a jejich vzdáleností. V nejnovější verzi programu je již navržena paměť, která uchovává informace o objektu pevně nastavený čas. To zamezuje ztrátě informací o pohybu objektu, který vstoupil za překážku. Když objekt opětovně vstoupí do viditelné oblasti, porovná se jeho výstřžek se všemi objekty v paměti. Pokud je nalezen podobný objekty, tedy poslední výstřžek před vstupem za překážku, přenesou se všechny informace z času před vstupem za překážku.

$$d = \sum_i^n \sqrt{(x_{1i} - x_{2i})^2 + (y_{1i} - y_{2i})^2} \quad (2)$$

4 VÝSLEDKY

Výsledkem práce je program, který detekuje potenciálně nebezpečné zavazadla na veřejných prostorech, kde tyto zavazadla mohou být nebezpečná svému okolí. Detekce je prováděna pomocí neuronové sítě pro detekci osob a zavazadel a implementovaného algoritmu pro sledování pohybu objektů a následné detekce vlastnictví zavazadla. Pokud je zjištěno, že zavazadlo nemá vlastníka a zároveň se nepohybuje, zavazadlo je na výstupním videu označeno červeným čtvercem. Tímto označením zavazadla dochází k upozornění obsluhy programu.



Obrázek 2: Výsledná detekce programu

5 ZÁVĚR

V této práci došlo k úspěšné implementaci programu, který je schopen detekovat potenciálně nebezpečná zavazadla. Program ještě stále není připraven na realné použití. Důvodem je chybějící implementace grafického uživatelského rozhraní a možnost připojit IP kameru. Tedy v den odevzdání práce do soutěže EEICT je velký prostor pro vylepšování programu.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu semestrální práce panu Ing. Václavu Uhrovi, za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

REFERENCE

- [1] REDMON, Joseph, Santosh DIVVALA, Ross GIRSHICK a Ali FARHADI. You Only Look Once: Unified, Real-Time Object Detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [online]. IEEE, 2016, s. 779-788 [cit. 2017-11-25]. DOI: 10.1109/CVPR.2016.91. ISBN 978-1-4673-8851-1. Dostupné z: <http://ieeexplore.ieee.org/document/7780460/>
- [2] How-To: Python Compare Two Images - PyImageSearch. PyImageSearch - Be awesome at OpenCV, Python, deep learning, and computer vision [online]. Copyright © 2017 PyImageSearch. All Rights Reserved. [cit. 29.11.2017]. Dostupné z: <https://www.pyimagesearch.com/2014/09/15/python-compare-two-images/>